Introduction to Technical Cybersecurity

This is a course map for ECE 529: An Introduction to Technical Cybersecurity. Here, we have modules broken down by topic matter, lessons, objectives, quizzes, and homework assignments. You should cover at least a module a week in order to keep up with the class. I suggest you give yourself two weeks per module for modules 6 and 7, and cover the preceding five modules in the remaining four weeks. This class is fast-paced, and it will be difficult to catch up if you fall behind. Furthermore the final modules are the most technically rigorous.

Introduction to Technical Cybersecurity	Modules	Lessons	Description	Work and Evaluation	Objectives
Module 1	**Cybersecurity: A History**	A general introduction to the state of malware development and execution today.			
		Cybersecurity Today	Describe the current cybersecurity landscape, touching in APT, ransomware, etc.		*(A) Explain the history of cybersecurity and hacking* *(B) Identify the key events socially and technically that lead us to where we are today.*
		The Beginnings	The origins of hacking, early malware of note, how they were developed, and why they worked.		
		The Middle Ages	The first involvement of criminal organizations, targets of these kinds of orgs.		
		The Path to APT	How we got to where we are today.	Quiz (summative)	
Module 2	**Architecture & Analysis**	Architectural issues around designing malware campaigns.			
		Campaign Architecture	This will cover the primary types of campaigns today, including ransomware, data exfiltration, extortion, espionage, and sabotage.		*(C) Explain the different types of cyber campaigns* *(D) Be able to defensively and offensively analyze a system*
		Attack Surfaces	Analyzing systems for pathways of attack. This section will cover threat modeling, attack modeling, and attack surface development.		
		Attack Surface Analysis	How to analyze a system from a defensive perspective. This will include goal analysis, risk analysis, and defensive controls.		
		The Cyber Killchain	How malicious actors design malware campaigns. This will include attacker goal setting and introduce the cyber kill chain and alternative perspectives (including DevOps).	Assignment: Design Campaign (summative)	
Module 3	**Reconnaissance & Vulnerability Identification**	Recon techniques and strategies used when exploring possible malware campaigns.			
		Search Engines	Google dorking is a thing. And it's not the only search engine available. How we can use various search engines to understand who does what in an organization.		*(E) Be able to (1) design and execute, and (2) defend against a malware campaign against a target*
		NMAP Background & Lab Configuration	Introducing NMAP and configuring a lab for upcoming work.		
		NMAP Scanning	Using technical means to recon a possible target. Includes information on Masscan, DNS recon, nmap, web recon techniques, and extracting information from various documents.	Assignment: Metasploit System (summative)	
Module 4	**Penetration & Delivery**	Typical delivery techniques and tools used for delivery. We'll discuss various techniques as well as toolsets like the Browser Exploitation Framework and Social Engineering Toolkit. Malware won't use these directly usually, but manual attacks may, and malware can use components from these toolsets.			
		Initial Exploitation	Campaigns that are based on remote exploitation attacks. While campaign development is architectural, this is more detailed campaign design work.		*(E) Be able to (1) design and execute, and (2) defend against a malware campaign against a target*
		Technical Details	Cracking tools, How ARP works, LANMAN & NTLM hashes		
		Malvertising, Exploit Kits, and Phishing	Campaigns designed around phishing, watering hole or malvertising techniques.	Assignment: Delivery System Design (summative)	
Module 5	**Binary Analysis**	Typical initial exploitation techniques. We'll start to look at metasploit here as well, and discuss metasploit components that would typically be reused in malware distributions. We'll also cover ways to organize personal exploit libraries in metasploit.			
		Binutils	Building a virtual workstation for binary program analysis and an introduction to GNU Binutils.		*(E) Be able to (1) design and execute, and (2) defend against a malware campaign against a target*
		GDB	Introducing GDB and showing how programs are really called.		

Introduction to Technical Cybersecurity

This is a course map for ECE 529: An Introduction to Technical Cybersecurity. Here, we have modules broken down by topic matter, lessons, objectives, quizzes, and homework assignments. You should cover at least a module a week in order to keep up with the class. I suggest you give yourself two weeks per module for modules 6 and 7, and cover the preceding five modules in the remaining four weeks. This class is fast-paced, and it will be difficult to catch up if you fall behind. Furthermore the final modules are the most technically rigorous.

Introduction to Technical Cybersecurity	Modules	Lessons	Description	Work and Evaluation	Objectives
		Instruction Architectures	Discussion of various instruction architectures like MIPS and ARM, how they differ from X86, and GDB command internals.	Assignment: Capture the Flag! (summative)	
Module 6	**Attacking the Stack**	This module covers techniques for malware persistence.			
		The Stack	An introduction to the call stack, how it works, and how it's used. We'll statically and dynamically analyze a call stack.		*(E) Be able to (1) design and execute, and (2) defend against a malware campaign against a target*
		Analyzing the Stack	Starting to look more in-depth at dynamic stack behavior using GDB and an exploitable program.		
		Stack Attack!	Attacking an exploitable program with a buffer overflow attack.	Assignment: Build exploitable program and exploit it (summative)	
Module 7	**Ret2libc & ROP**	We've finished with buffer overflows. Here, we'll look at attacks that were developed after defenses against buffer overflows were put into place.			
		Ret2Libc	Return to libc attacks came into vogue after buffer overflow attacks were blocked. We'll learn how they work here.		*(E) Be able to (1) design and execute, and (2) defend against a malware campaign against a target*
		Return Oriented Programming	Attackers developed return oriented programming (ROP) after address space in programs were randomized. Here we discuss how it works and what it is.		
		Using ROP	We've covered ROP, now let's look at how we could use it.	Assignment: Using Ret2libc	
Module 8	**Lab Week**	We've covered lots of material! This is an extra week to submit work you're not happy with or you've fallen behind on.			